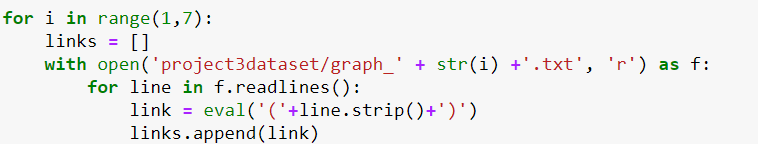
Link Analysis Practice

E94056178 工科系 盧柏翔

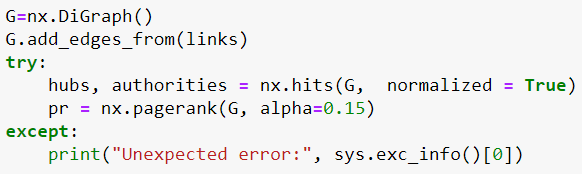
* **Implementation(詳見main.ipynb)**

Task 1: Please implement HITS and PageRank

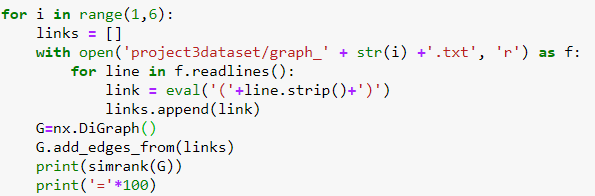
STEP1:讀檔並建立連結存至list



STEP2-1:利用python的networkx套件分別計算各Graph的hub、authority、pagerank值



STEP2-2:計算前5 Graph的SimRank值

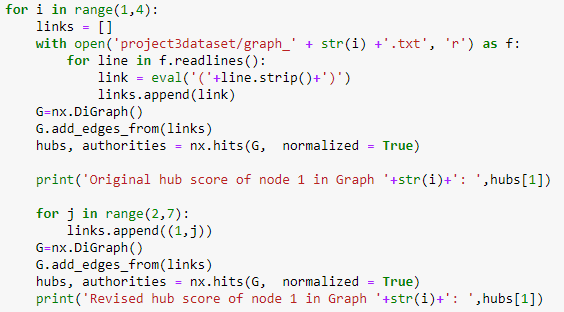


SimRank 函式實作

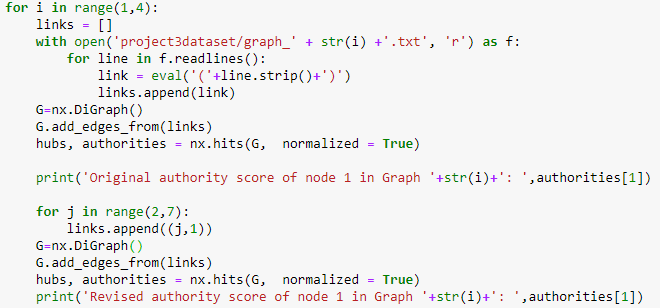


Task 2: Find a way to increase hub, authority and PageRank of Node 1 in first 3 graphs respectively

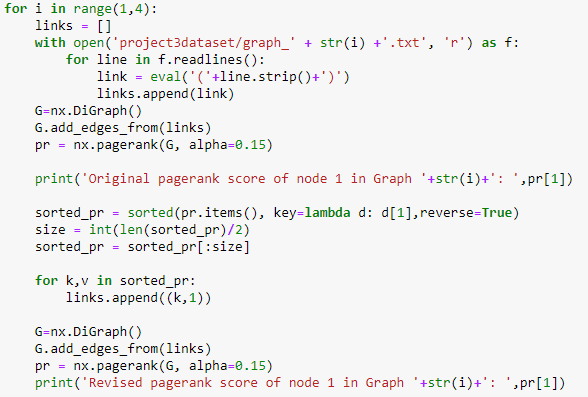
1. 提高Node1 hub值(想法:增加Node 1到其他點的link)



1. 提高Node1 authority值(想法：增加其他點到Node 1的link)

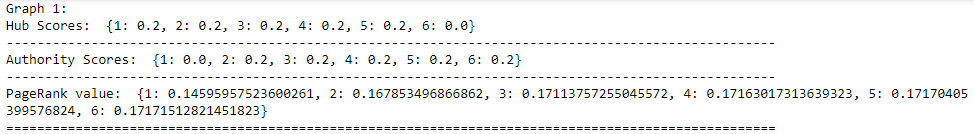
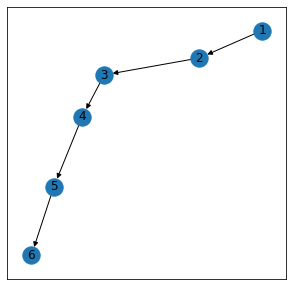


1. 提高PageRank 值(想法：盡量找PR值比較高且外部連結又少的點)



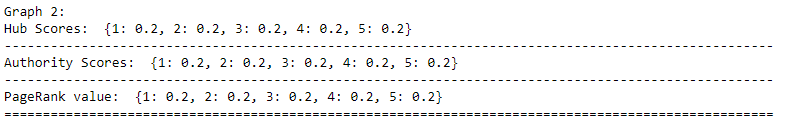
* **Result analysis and discussion**

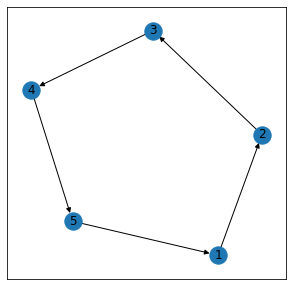
**各graph的hub,authority 和 pagerank的值**

1. **Graph 1**

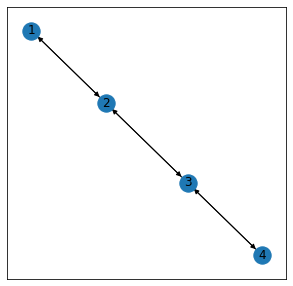
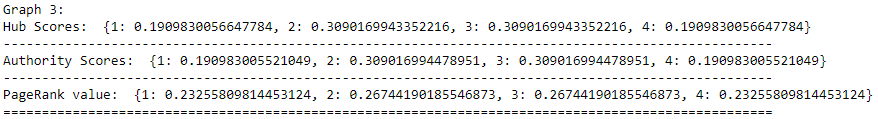
分析：Graph1 圖為單向一直線，可以注意到在Node1因無parent node所以authority為0，Node6因無chlid node 所以hub為0，使用PageRank則可以避免此現象，因演算法賦予每個頁面最小值。

****

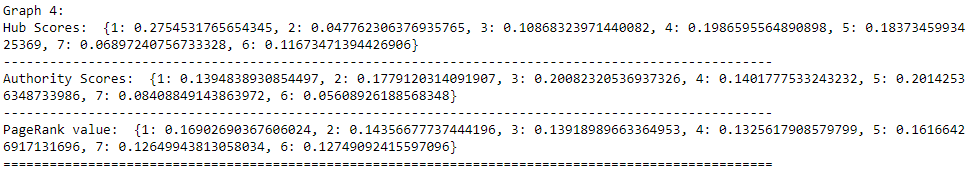
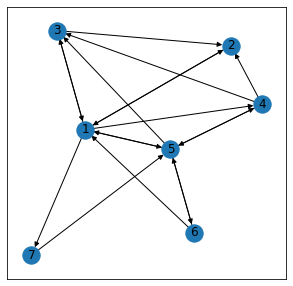
1. **Graph 2**



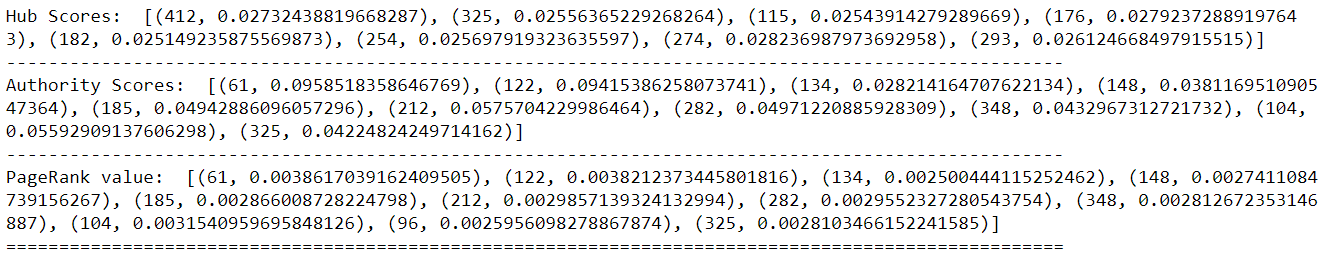
分析：Graph2 圖為單向環狀，導致hub,authority和pagerank值的結果跟初始值一樣並未改變。Hub值為child的authority相加再歸一化，Authority值為parent的hub相加再歸一化，pagerank則像水流般傳值，每個點的前後狀況均相同，因此值均一樣。

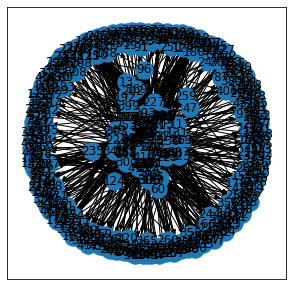
1. **Graph 3**

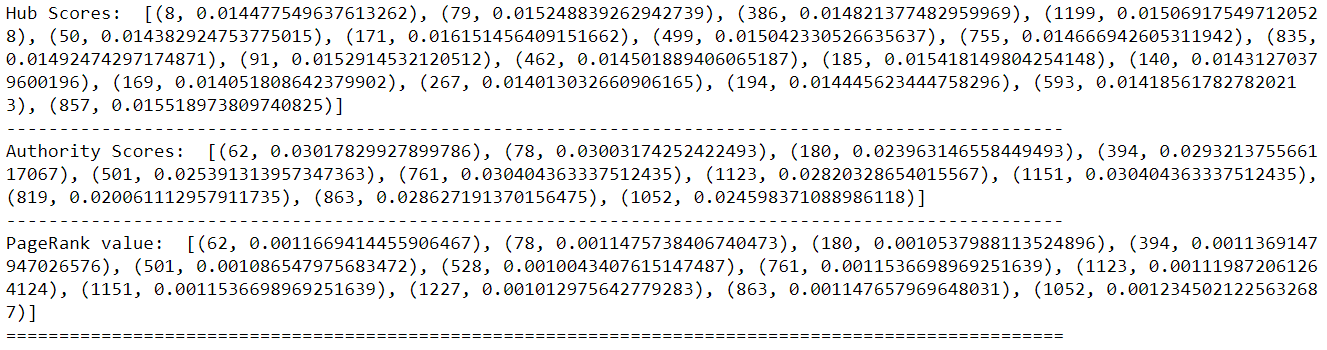
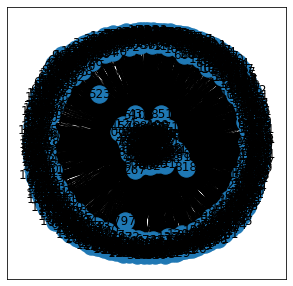
分析：Graph 3圖為直線雙向連結，與graph1只有單向不同，所以各點均有值，但node2和node3，indegree和outdegree 比1與4多，總的來說三個值都會比較高。

1. **Graph 4**

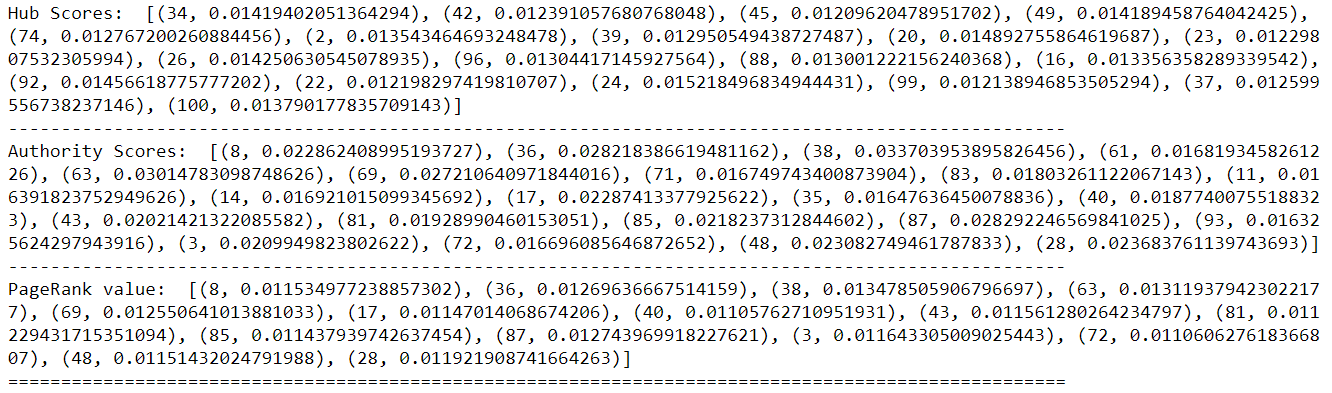
分析：從圖可看出重要的node像是node5和node3因為被許多node指向所以有較高的authority值，而node 1因為指向node3和node5較好的node，所以node1的hub很高。而node5的pagerank高因為被好的node連結(像是node1)，且連接出去的link少。

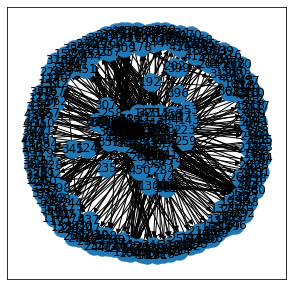
1. **Graph 5**

分析：此graph較複雜，有需多node但是大部分的node並沒有太多的Indegree和Outdegree，在weights初始均分的情況下，大部份的值都在重要少數的node上，造成其他node的hub,authrotiy,pagerank都趨近為零。此處列出前幾個值較大的node，發現hub,authority,pagerank的node都差不多。

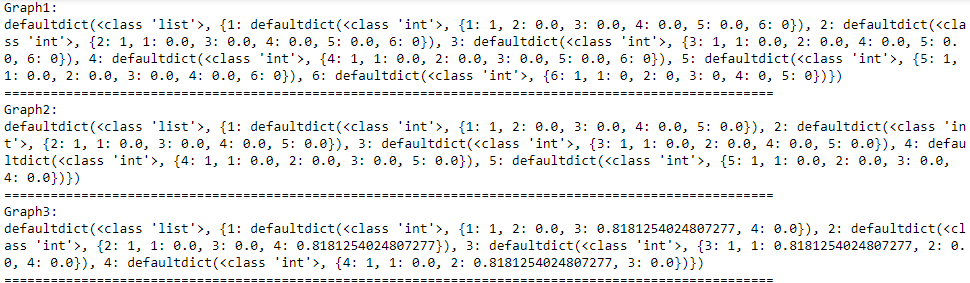
1. **Graph 6**

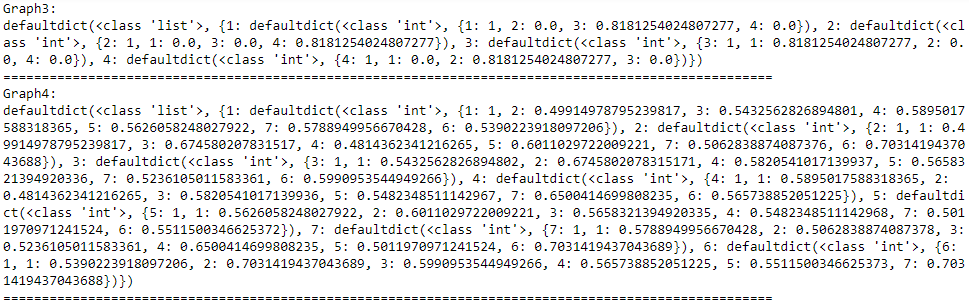
分析：此graph最複雜，約進行了120次iteration才達到收斂，收斂的結果跟graph5情況相似，大部分的node的hub,authority,pagerank都趨近零。此處列出前幾個值較大的node，發現hub,authority,pagerank的node都差不多。

1. **Graph from transaction data**

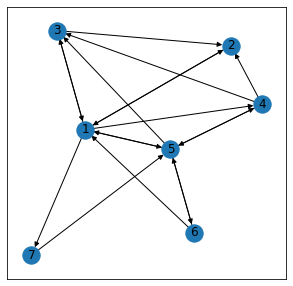
分析：由transaction data產生的graph，大部分的node未與其他node連接，因此跟前兩個graph相似，node多導致值小且集中在少數node上。

**SimRank(Graph5太大不列，詳見main.ipynb)**

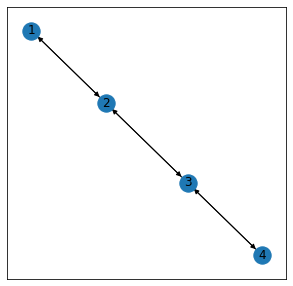
****

****

Simrank是評估任意兩節點相似度的演算法，主要思想為兩個物體與相似物體有關聯，那麼這兩個物體是相似的。通過結點之間的結構化的相同點來計算結點之間的相似度。



以左圖Graph4為例，因node6與node7均有指向node5所以相似度較高為0.703。



以左圖Graph3為例，node1和node3均指向與被指向node2相似度高為0.818；同樣node2和node4也是相同情形相似度為0.818。

**Node1增加hub, authority和pagerank**

提升hub主要思想為盡量將自身網站連向具有高authority的網站，因此實作主要將node1連接出去；提升authority主要思想為盡量將具有高hub的網站連向node1，因此實作主要將其他node連接進來；提升pagerank主要思想為盡量讓pagerank值高的網站連向node1，因此實作主要將前幾個高pagerank的node連向node1。

* Computation performance analysis

Graph越複雜耗時越久

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HITS** | Graph1 | Graph2 | Graph3 | Graph4 | Graph5 | Graph6 | Graph7 |
| Time(sec) | 0 | 0 | 0 | 0.001 | 0.053856 | 1.368556 | 1.108543 |
| iteration | 2 | 2 | 11 | 20 | 19 | 120 | 73 |
|  |  |  |  |  |  |  |  |
| **PageRank** | Graph1 | Graph2 | Graph3 | Graph4 | Graph5 | Graph6 | Graph7 |
| Time(sec) | 0 | 0.000997 | 0 | 0.001 | 0.007978 | 0.024933 | 0.039893 |
|  |  |  |  |  |  |  |  |
| **SimRank** | Graph1 | Graph2 | Graph3 | Graph4 | Graph5 |  |  |
| Time(sec) | 0 | 0 | 0.000997 | 0.00598 | 3.933138 |  |  |
|  |  |  |  |  |  |  |  |

* Discussion

More limitations about link analysis algorithms

Answer:

Hits的限制─依照邏輯，你不需要是位優秀的作家，才能評論其他作家，你只要是一位備受肯定的評論家就行了。HITS算法還存在易被作弊者操縱結果，結構不穩定等問題。

Pagerank的限制─沒有區分網站內的相互連結、沒有過濾廣告連結和功能連結、對新網頁不利。

Can link analysis algorithms really find the “important”

pages from Web?

Answer: 並不一定，若是存在許多垃圾網頁指向某網域，可能會造成某網頁是重要的假象，而且常常會受一些廣告連結干擾。

What are practical issues when implement these

algorithms in a real Web?

Answer: 實際各個網頁是動態的，不容易收斂變穩定，且太多的廣告連結會干擾分析效果。pagerank在實際應用中，Web連接圖中常常存在一些indegree或outdegree為0的node，這時會出現兩種異常：等級變化（RankLeak）和等級下沉（RankSink）

Any new idea about the link analysis algorithm?

Answer: 先人工決定較好或較信任的網站，信任度最高，之後向外連接逐漸下降，主要思想是好的網站很少連到壞的網站，但相反不成立，希望藉此找出更多好的網站。

What is the effect of “C” parameter in SimRank?

Answer: SimRank公式如下，參數C像是阻尼係數，可以這樣理解：假如I（a）= I（b）= {A}，按照（1）式計算出sim（a，b）= C \* sim（A，A）= C，所以C∈（0， 1），C的大小影響Simrank value 高低。

